In the Claims:

Please amend the above-identified application as follows:

Claim 1 (currently amended) A method of designing logical circuits, comprising the steps of:

a. representing ~~the~~ logic of a logical circuit to be designed as points and vectors in a vector space; and

b. using the points and vectors in ~~a~~ said vector space to simplify the logic of the logical circuit to a simpler form; and

c. designing the logical circuit using the simpler form.

Claim 2 (currently amended) A method of manufacturing logical circuits, comprising the steps of:

a. representing ~~the~~ logic of a logical circuit to be manufactured as points and vectors in a vector space; and

b. using the points and vectors in ~~a~~ said vector space to simplify the logic of the logical circuit to a simpler form; and

c. using the simpler form to implement the logical circuit in hardware.

Claim 3 (original) A method of simplifying logical circuits, comprising the steps of:

a. representing the logic of a logical circuit as points and vectors in a vector space; and

b. modifying the representation in vector space using at least one process rule of a set of process rules to simplify the logic.

Claim 4 (currently amended) The method of claim 3 in which at least one process rule of a set of process rules consisting of one of the following process rules:

a. Process Rule 1-- comprising the steps of

~~a1.~~ Representing in alternational normal schema, ~~the~~ a target schema **t**, as a set of vectors in ~~the~~ ANS-space,

~~a2.~~ Represent ~~E~~each clause or disjunct of **t** ~~is~~ as a position vector ~~(i.e. one~~ pointing to **O**~~)~~ with **O** at one corner of a set of parallelograms made of propositional addresses to ~~the~~ an ί-point at ~~the other~~another corner, and

~~a3.~~ Using ~~A~~any two other outside vertices of such a parallelogram ~~are~~as implicants which are among ~~the~~ original clauses of **t**; and

b. Process Rule 2-- comprising the steps

~~b1.~~ Picking any two clauses, and

~~b2.~~ If there is a propositional address σ at ~~the~~a midpoint between ~~the~~ component

clauses, ~~the~~a vector from ί to σ, is ~~the~~a simplification of and can replace ~~the~~ relevant clauses ~~of~~; and

c. Process Rule 3~~—~~comprising the steps of

~~c1.~~ Generateing ί-implicants until each clause or vector has been used at least once, and

~~c2.~~ If a disjunct **d** of **t** cannot be used because it forms no propositional address with any other disjunct, then **d** must appear unmodified in ~~the~~a final schema which is the simplification of **t**; and

d. Process Rule 4~~—~~comprising the step of

~~d1.~~ If an ί-point exists in **t**, deleteing the vectors which produce it in favor of the vector from ί to **O**; and

e. Process Rule 5~~—~~ comprising the step of

~~e1.~~ For a clause in a schema which subsumes another clause eliminateing ~~the subsuming~~ said clause which subsumes; and

f. Process Rule 6~~—~~ comprising the step of

~~f1.~~ Couples such as **pq v** $\bar{p}\,\bar{q}$ or $\bar{p}\,\bar{q}$**s v pq**$\bar{s}$ cannot be summed to zero at the origin; and

g. Process Rule 7~~—~~ comprising the steps of

~~g1.~~ Translateing vectors if a corresponding σ-point exists for an ί-point then σ is the simplification of ί, and

~~g2.~~ Indentifying ~~A~~any superpositions of parallel arrows in opposite directions ~~represent~~as equivalences, and

~~g3.~~ For equivalences, (a) drop the longer clause at either end of any double-headed arrow, (b) drop pairs, triples etc. of double-headed arrows which meet at a point in favor of the vector from that point to **O** and (c) drop a vector or clause in the target schema which is itself the resultant of any other two vectors; and

h. Process rule 8~~—~~ comprising the steps of

~~h1.~~ ~~A~~Determining simplification is complete if in ~~the~~a representation ~~system~~of simplified layers which replaces the target schema no vectors or clauses are subsumed by others and no double-headed vectors remain.

Claim 5 (currently amended) Apparatus for simplifying logical circuits, comprising:

a. a processing element configured to represent ~~the~~ logic of a logical circuit to be simplified as points and vectors in a vector space and to use the points and vectors to simplify the logic of the logical circuit to a simpler form.

Claim 6 (original) The apparatus of claim 5 in which the processing element is an optical computer.

Claim 7 (original) The apparatus of claim 5 in which the processing element is a digital computer.

Claim 8 (currently amended) The apparatus of claim ~~1~~5 in which the processing element is an colorimetric computer.

Claim 9 (currently amended) The apparatus of claim ~~1~~5 in which the processing element is an analog computer.

Claim 10 (currently amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing ~~the~~ logic of a logical circuit to be designed as points and vectors in a vector space and for using the points and vectors in a vector space to simplify the logic of the logical circuit to a simpler form and for designing the logical circuit using the simpler form.

Claim 11 (currently amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing ~~the~~ logic of a logical circuit to be manufactured as points and vectors in a vector space, and for using the points and vectors in a vector

space to simplify the logic of the logical circuit to a simpler form, and for using the simpler form to implement the logical circuit in hardware.

Claim 12 (currently amended) A computer program product, comprising:

a. a memory element; and

b. a computer program stored on said memory medium, said computer program comprising instructions for representing ~~the~~ logic of a logical circuit as points and vectors in a vector space, and for modifying the representation in a vector space using at least one process rule of a set of process rules to simplify the logic.